

A DISTRIBUTED SYSTEM FOR LARGE-SCALE GEOLOCALIZATION OF INTERNET HOSTS

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of
Master of Science

by

Nicole Lee Caruso

January 2011

© 2011 Nicole Lee Caruso
ALL RIGHTS RESERVED

ABSTRACT

Determining the geographic location of nodes enables a variety of useful network applications. Such applications include finding servers that are closest to clients, customizing content based on the location of website visitors, locating faulty nodes during network diagnosis, and tracking down persons of interest. Past efforts to solve the geolocalization problem have either relied on databases of questionable accuracy and integrity or on limited sources of information about node location. As a result, their coverage and accuracy have suffered.

This thesis presents Alidade, a distributed framework for performing accurate and scalable geolocalization with worldwide coverage. Alidade provides a principled approach to geolocalization based on geometric constraint satisfaction. It takes advantage of diverse sources of ground truth, including landmarks whose positions are approximately known. The framework partitions constraints extracted from the network into disjoint sets that can be independently and concurrently evaluated. This enables a MapReduce style implementation that scales well with the number of landmarks and targets.

BIOGRAPHICAL SKETCH

Nicole Lee Caruso received her undergraduate education at Rutgers University, joining the Honors Engineering Program with a Presidential Scholarship. In May 2008, she graduated with a B.S. in Electrical and Computer Engineering as a James J. Slade Scholar, earning the Potter Award for her undergraduate research work. With Kuang Sheng, she worked on a JFET-based DC-DC converter by designing the triangle waveform generator and bandgap reference circuit. With Yanyong Zhang at Winlab, she helped implement “R-Sentry”, which provides continuous sensor services against random node failures. Nicole received her graduate education at Cornell University. In January 2011, she graduated with a Master of Science in Electrical and Computer Engineering. With Emin Gün Sirer, she played a lead role in developing “Alidade”, a distributed system for large-scale geolocalization of internet hosts. In addition, Nicole worked on “HyperDex”, a distributed multidimensional key-value store for cloud computing. She currently lives in the Silicon Valley area.

This thesis is dedicated to my family.

ACKNOWLEDGEMENTS

This work evolved with the help of Emin Gün Sirer and Bernard Wong at Cornell University, Bruce Maggs at Duke University, and researchers at Akamai Technologies.

TABLE OF CONTENTS

Biographical Sketch	iii
Dedication	iv
Acknowledgements	v
Table of Contents	vi
List of Figures	vii
1 Introduction	1
2 Framework Foundation	4
2.1 Network Database	4
2.1.1 Active Landmarks	4
2.1.2 Beacons	5
2.1.3 Intermediate Nodes	7
2.2 Geometric Constraints	7
2.3 Constraint Extraction	9
2.3.1 Speed-of-Light	9
2.3.2 Probabilistic Techniques	10
2.4 Constraint Accumulation	13
2.5 Distributed Framework	15
3 Further Enhancements of Framework	18
3.1 Intermediate Node Localization	18
3.2 Intermediate Node Latencies	18
3.3 Public Host Extraction	20
3.4 Additional Geolocalization Tools	22
3.5 Geopolitical Boundaries	24
3.6 Geodesic-Aware Projection	26
3.7 Region Representation	28
3.8 Region Expansion	28
3.9 Region Boundary Crossing	30
4 Evaluation	32
5 Related Work	36
5.1 Single Point Geolocalizaion	36
5.2 Region Geolocalization	37
5.3 Wireless Geolocalization	39
6 Conclusion	40
Bibliography	41

LIST OF FIGURES

2.1	Distribution of PlanetLab Hosts Locations	5
2.2	Distribution of Akamai Server Locations	6
2.3	Geometric Constraint Concepts	7
2.4	Latency-to-Distance Mapping and Constraints	8
2.5	Latency-to-Distance Constraints for Probabilistic Methods	9
2.6	Extracted Geometric Constraints	10
2.7	Weighted Intersection Example	14
2.8	MapReduce Process for Target Localization	16
3.1	Beacon Demonstration	19
3.2	Traceroute Example	20
3.3	Localization Accuracy of Public Hosts	21
3.4	Distribution of public host Locations	22
3.5	City Name to Region Mapping in Octant	23
3.6	City Name to Region Mapping in Alidade: United States	24
3.7	City Name to Region Mapping in Alidade: South Korea	24
3.8	Convex Hull Representation of Geopolitical Boundaries	25
3.9	Flat World and Geodesic-Aware Approximations	26
3.10	Efficient Area Arithmetic	27
3.11	Area Expansion Techniques	29
3.12	Example Cases for Area Expansion in Alidade	30
4.1	Evaluation of Localization Accuracy	33
4.2	Evaluation of Localization Precision	34
4.3	Evaluation of Components on Localization Metrics	35

CHAPTER 1

INTRODUCTION

Geolocalization, that is, determining the geographic location of nodes from network measurements, is an important building block for many applications. Such applications include finding servers that are closest to clients, customizing content based on the location of website visitors, locating faulty nodes during network diagnosis, and tracking down persons of interest. Many of the previous approaches to geolocalization have been ad hoc, relying on databases of questionable accuracy, integrity, coverage, and maintenance. Other approaches have been very conservative, relying on very lax constraints and limited sources of ground truth. As a result, their accuracy and coverage have suffered. On the other hand, localizing a target based on landmarks and network latency measurements provides high integrity and helps eliminate unexpected drastic errors.

This thesis presents Alidade, a distributed framework for performing accurate, scalable, low-latency geolocalization with worldwide coverage. Alidade derives a location estimate of a target node based on hosts with known locations, called landmarks, and latency measurements between these landmarks and the target. Alidade takes advantage of many diverse sources of ground truth, including nodes whose geographic locations are precisely known, and even nodes whose locations are only approximately known. The framework uses this information to provide a principled approach to geolocalization based on geometric constraint satisfaction.

Alidade’s objective is to extend and surpass its ancestor, Octant [16], our previous work for geolocalization with geometric constraints. Previously, Octant

had a restricted scope that was limited to several dozen nodes within the continental United States. Alidade’s goal is to be able to localize targets that exist anywhere throughout the globe. Alidade builds on some of the techniques introduced in Octant. This thesis discusses several new fundamental components that come into play in order to achieve worldwide geolocalization.

The framework achieves high accuracy for targets distributed across the world by utilizing all available sources of ground truth. A concentrated and widespread collection of landmarks helps improve the accuracy and precision of the localization. Constituting the collection of hosts with known locations is PlanetLab hosts, as well as a global-scale dataset of Ecor web servers from Akamai Technologies, which was not available in Octant or other previous geolocalization approaches. This information is supplemented with approximate locations of select routers from the UNDNS [15] and HostParser¹ tools as well as a dataset of geopolitical boundary constraints. Alidade can further integrate other sources of ground truth, including hosts listed on a public mapping database such as Google or Bing local search.

Alidade can efficiently combine positive and negative constraints, even when the positions of the landmarks from which they are derived are imprecise. Alidade uses a compact and precise method to represent a geographic region, which can be either an individual geometric constraint or a feasibility region containing the set of points at which a target is estimated to reside. A region is expressed as an area bounded by straight-line segments, and may be non-convex, be disconnected, or contain holes. This enables the system to not only reason about positive constraints (i.e. information about where a node is likely to be located), but also negative constraints (i.e. information about where

¹HostParser derives much inspiration from its ancestor, EdgeScape [1].

a node is *not* likely to be). This representation enables quick and space-efficient computation of geometric intersection, union, and subtraction. In addition to these functions, Alidade introduces an efficient geometric expansion technique that dilates the feasibility region of an approximately known landmark based on latency information. The result is a new geometric constraint for a target using a landmark whose location is only approximately known.

For global-scale geolocalization, runtime would be astronomical unless parallelism is introduced. Alidade is a distributed framework that is inherently parallel and scales to extensive landmark and latency datasets and geometric computations. In several minutes, the framework can handle thousands of landmarks and perform millions of geometric computations. Alidade leverages the availability of cheap commodity machines in order to reduce the runtime significantly. It is constructed as a multistage MapReduce [8] process, where each stage increasingly improves accuracy and precision. The framework partitions constraints extracted from the network into disjoint sets that can be independently and concurrently evaluated. This enables a MapReduce style implementation that scales well with the number of landmarks and targets by efficiently utilizing the inherent parallelism in the underlying geolocalization computations. The worldwide geolocalization that Alidade accomplishes in less than ten minutes does not reach completion on the Octant framework.

CHAPTER 2

FRAMEWORK FOUNDATION

Alidade aims to accurately and precisely localize Internet hosts that are anywhere on the globe. There are several key concepts that are fundamental to Alidade’s approach to geolocalization.

2.1 Network Database

Alidade’s main sources of information include landmarks and network measurements. Active landmarks initiate traceroutes. A target node is an Internet host whose location is unknown; determining its location is the objective of geolocalization. A landmark is an Internet host whose location is known; it anchors the system to real world coordinates.

2.1.1 Active Landmarks

Active landmarks initiate traceroutes to targets in order to collect connectivity and latency data. Alidade uses PlanetLab, a shared research test bed with hundreds of nodes distributed across the globe, to provide a set of active landmark nodes. PlanetLab hosts have precisely known locations and can both initiate and respond to traceroutes. Figure 2.1 illustrates the worldwide distribution of PlanetLab hosts. Alidade has found 1,013 PlanetLab hosts in 381 unique locations.

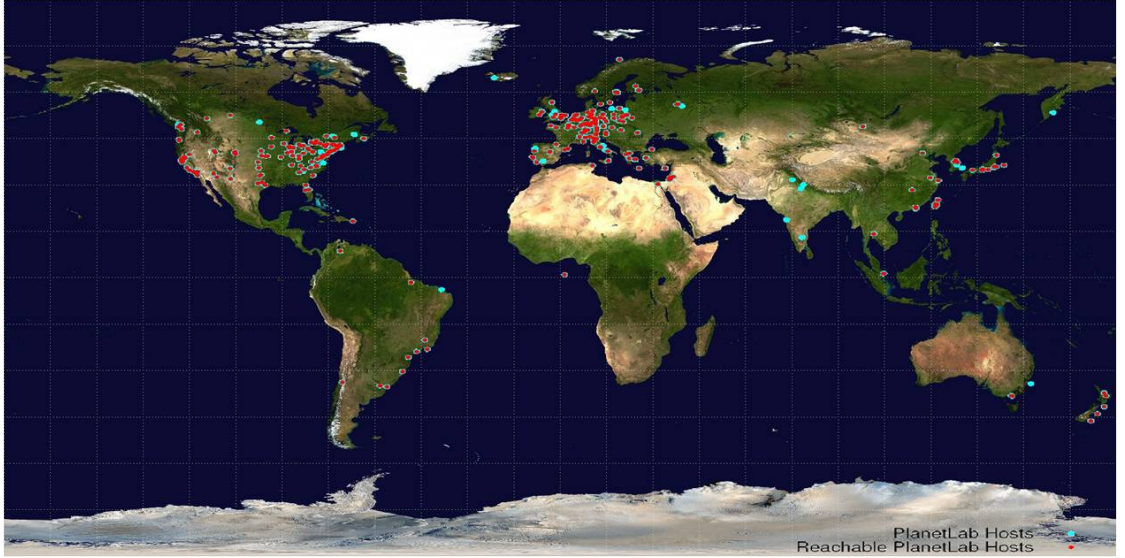


Figure 2.1: Distribution of PlanetLab host locations.

2.1.2 Beacons

To further improve the accuracy and precision of localizations on a global scale, Alidade leverages a large-scale dataset of beacon nodes. Beacons are passive landmarks that cannot initiate traceroutes, but are responsive to them. Thus, the system can gather intermediate node latency measurements and connectivity information from traceroutes to these beacons. Suppose an intermediate node is on both a traceroute path from a landmark to a target, as well as a traceroute path to beacon. The system can use the beacon to localize the intermediate node, and in turn use this intermediate node to localize the target more precisely. For a detailed example, see Section 3.1.

The beacon set includes a global-scale database of Ecor web servers from Akamai Technologies, which was not available in Octant or other previous geolocalization approaches. Akamai servers have precisely known locations and

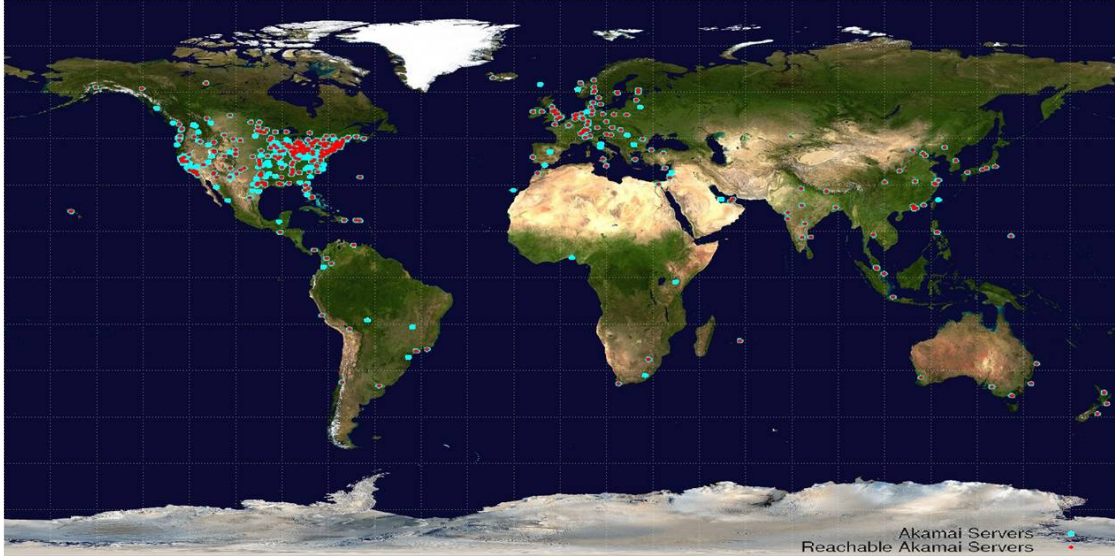


Figure 2.2: Distribution of Akamai server locations.

can respond to traceroutes. We cannot initiate traceroutes from them because they are live production servers. Figure 2.2 illustrates the worldwide distribution of Akamai servers. At the time of writing, there exist 29,840 Akamai servers in 650 unique locations.

Alidade uses the Bing Phonebook database to expand the set of beacons with publicly available host information. Alidade has a large database of public hosts, where website domains serve as landmark IP addresses and street addresses serve as known locations. Hosts are listed on public mapping databases such as Google or Bing local search. For example, restaurants and colleges make their addresses and websites publicly available. For information about the extraction of public hosts, see Section 3.3.

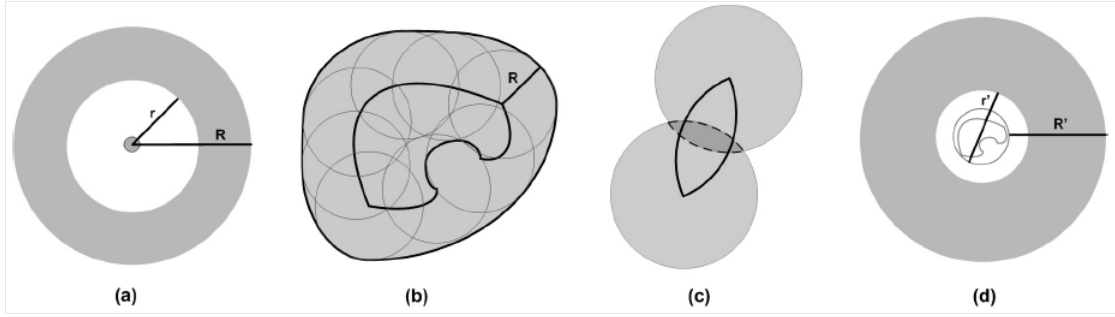


Figure 2.3: Geometric constraint concepts. (a) Positive and negative constraints from a landmark with a precisely known location. (b) Positive constraints from a landmark with an approximately known location. Only a sample of circles is shown for clarity. (c) Negative constraints from a landmark with an approximately known location. Only a sample of circles is shown for clarity. (d) Positive and negative constraints from a landmark whose feasibility region is replaced by a bounding circle.

2.1.3 Intermediate Nodes

Intermediate nodes are passive landmarks that are not end hosts of traceroutes. Information about these intermediate nodes further enhances the accuracy and precision of the system. This group consists of routers, whose locations are approximately known, either from previous localizations or other geolocalization tools, such as UNDNS and HostParser.

2.2 Geometric Constraints

Alidade approaches geolocalization as a system of geometric constraints. A set of landmarks, together with traceroute data from and to these nodes, produce

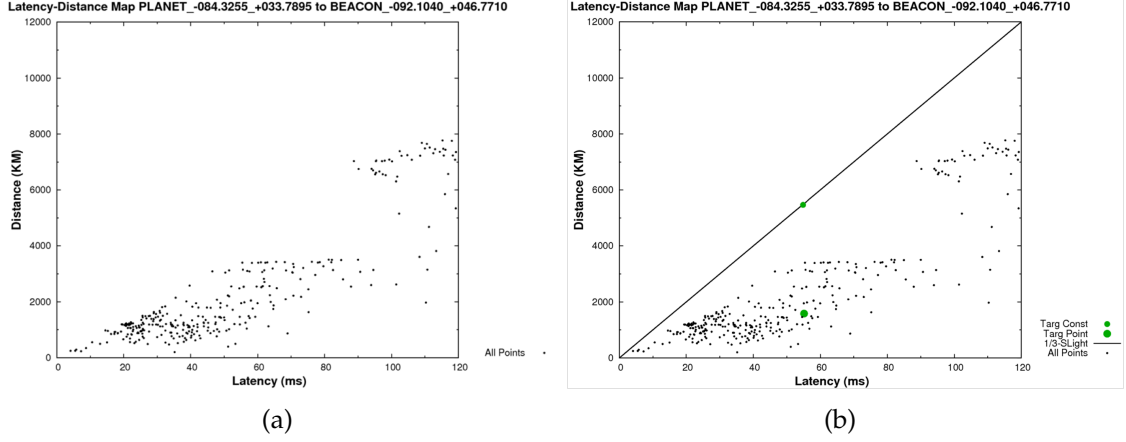


Figure 2.4: Latency-to-Distance mapping and constraints. (a) Mapping of network latency measurements. (b) Speed-of-Light method.

geometric constraints that describe where a target likely resides. Geometric constraints indicate the regions where a target is likely to exist. Figure 2.3 demonstrates comprehensive use of positive and negative constraints from various types of landmarks. For a landmark with a precisely known location, positive and negative constraints form a ring-like region. For a landmark with an approximately known location, the system calculates positive constraints by taking the union of all circles in the landmark's feasibility region. A target likely resides in the shaded region. The system calculates negative constraints by taking the intersection of all circles in the landmark's feasibility region. A target likely resides outside of the dotted line. Alidade can replace a complex feasibility region with a simple bounding circle. This forms a conservative constraint that is a superset of the constraint that would be derived from the original feasibility region.

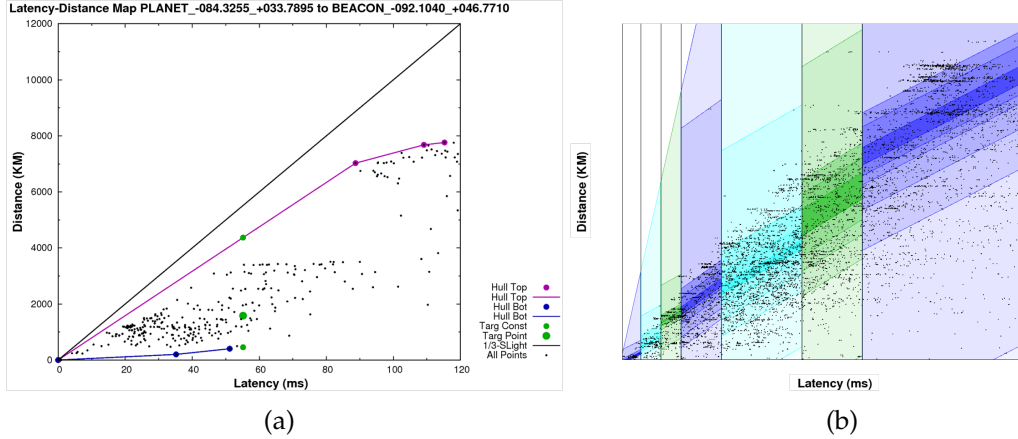


Figure 2.5: Latency-to-distance constraints of probabilistic methods. (a) Convex Hull method. (b) Density Distribution method.

2.3 Constraint Extraction

Observing that there exists a positive correlation between latency and distance, these constraints can arise from network latency measurements. Network latency measurements form a latency-to-distance mapping, where each latency is associated with a distance that indicates the size of the geometric constraint region. The system can derive constraints from either an individual latency-to-distance mapping for each landmark or a cumulative latency-to-distance mapping for all landmarks. Alidade introduces a number of methods that map network latency measurements to distance constraints.

2.3.1 Speed-of-Light

The Speed-of-Light method, illustrated in Figure 2.4(b), is a very conservative approach that uses the physical limit of wired transmission speed (two-thirds the speed of light on copper wire and on fiber optic cables) to associate latency

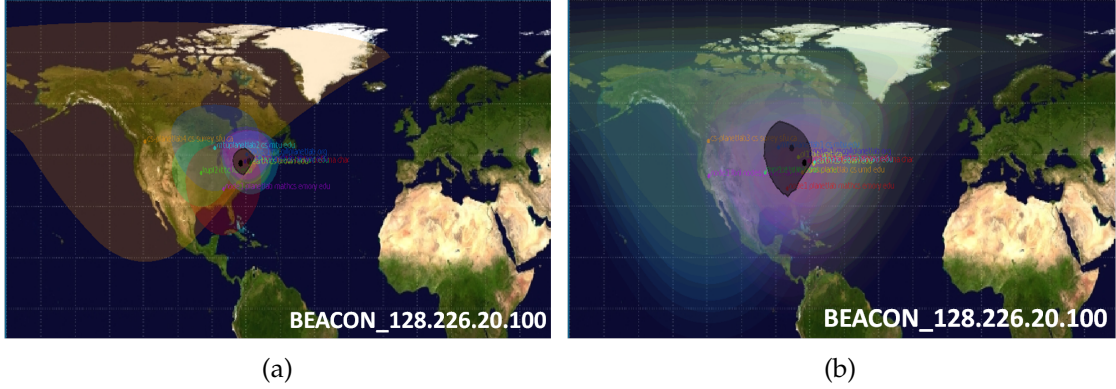


Figure 2.6: Extracted geometric constraints. (a) Convex Hull method. (b) Density Distribution method.

with distance. It assumes that distance is at most linearly related to latency based on this constant, and thus bounds the feasibility region above by the propagation speed. The Speed-of-Light method ensures that each constraint region contains the target with absolute certainty, and thus eliminates localization regions that do not contain the target.

2.3.2 Probabilistic Techniques

Speed-of-Light constraints are at times excessively conservative in that they do not provide sufficient localization precision. This is especially true for targets that have few network latency measurements or only have information from landmarks that are far away. For such a case, extracting constraints is non-trivial in that latency and distance, in practice, are not linearly related by a physical constant. Instead, latencies are often erratic and dilated due to network factors such as circuitous routing, queuing delays, and software delays.

To localize a node with superior accuracy and precision, Alidade uses con-

straint extraction methods that fully utilize the multitude of collected network information and construct more statistically stable latency-to-distance mappings. With Alidade’s worldwide large-scale network database of active landmarks, beacons, and intermediate nodes, the system can gather a sufficient number of data points to interpolate trends from collected traceroute data and derive statistical distributions to aid in constraint extraction. The system has three novel constraint extraction techniques: the Convex Hull method, the Density Distribution method, and the Kernel Distribution method.

Convex Hull

The Convex Hull method, illustrated in Figure 2.5(a), is an aggressive approach that creates narrow distance constraints. It uses both positive and negative area constraints, or areas that describe where a node must be located and areas that describe where a node cannot be located, respectively. This method ensures that constraints do not violate previous empirical measurements. The Convex Hull envelopes all points on the latency-to-distance mapping, where the top portion represents the upper bound (outer radius) and the bottom portion represents the lower bound (inner radius). All points outside the Convex Hull are considered infeasible, thus producing areas that represent negative constraints. The uncertainty of the latency-to-distance mapping increases with latency, since the density of data points, as well as the number of landmarks associated with these data points, tends to taper off. Thus, the system relaxes constraints as latency increases by ensuring that the top and bottom constraints never intersect. Figure 2.6(a) illustrates the extracted geometric constraints.

Density Distribution

The Convex Hull method assumes that the probability that a target exists slightly within the Convex Hull is one, while the probability that a node that exists slightly outside the Convex Hull is zero. To avoid such an abrupt transition, Alidade can use smoother statistical distributions to extract constraints. Alidade's large-scale network database provides many data points that enable the derivation of statistical distributions with much smoother probability transitions. In Octant, these elaborate constraint techniques were not feasible due to limited measurements and computational power.

The Density Distribution method partitions the latency-to-distance mapping into discrete regions, illustrated in Figure 2.5(b), such that bin partitions are drawn in parallel with the y-axis and step partitions are drawn in parallel with the linear least squares (best fit) line for each bin. The locations of bin partitions are placed so that each bin contains the same number of data points. Likewise, within each bin, the locations of step partitions are placed so that each step contains the same number of points. The weight of a geometric constraint is inversely proportional to the size of the associated partition in the latency-to-distance mapping. In other words, the weight assignment is a function of the density of latency-to-distance data points. Recalling that the uncertainty of the latency-to-distance mapping increases with latency, the Density Distribution naturally widens the upper and lower steps as latency increases, where the upper bound cutoff is two-thirds the speed of light. Figure 2.6(b) illustrates the extracted geometric constraints.

Kernel Distribution

The Density Distribution method has discontinuities among bin partitions that abruptly affect geometric constraint weights depending on which bin the target falls into. On the other hand, the Kernel Distribution method partitions the latency-to-distance mapping with continuous partitions that separate regions based on the probabilities associated with them. The Kernel Distribution method is inspired by [17]. A relatively smooth two-dimensional Kernel-based distribution in three-dimensional space is generated from traceroute information. This method bases each weight assignment on the Kernel-based probability that a target node can be found at a particular distance from a landmark, given a latency value. In other words, the weight of a geometric constraint is a function of the Kernel-based probability on the latency-to-distance mapping.

2.4 Constraint Accumulation

Alidade integrates individual constraints in order to arrive at a target’s final localization region. For Speed-of-Light constraints provided by active landmarks with precisely known locations, Alidade performs simple Boolean calculations on the areas. Specifically, it takes the intersection of all positive constraints to arrive at a cumulative area that all individual constraints agree upon. This is equivalent to taking the union of all negative constraints, where the remainder is the target’s feasibility region.

All other types of constraints have the potential to erroneously exclude the target’s actual location. Therefore, Alidade computes the weighted intersec-

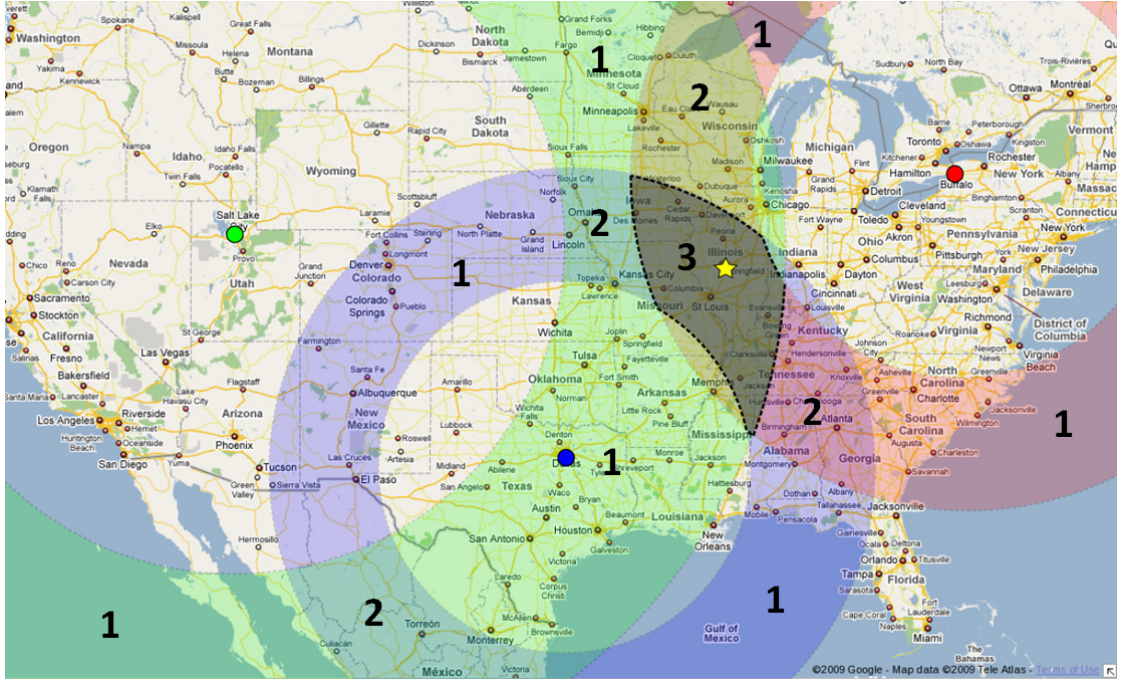


Figure 2.7: Example of a weighted intersection that uses three landmarks (dots) to localize a target (star).

tion of these constraints, illustrated in Figure 2.7. Each constraint is assigned a weight based on the latency, the type of landmark, and the type of information the landmark’s feasibility region is based on. The weight is inversely proportional to latency. This approach helps prevent null localizations due to overly aggressive constraints and provides more flexibility to achieve a desired precision, or region size.

The union of all weighted increments that satisfy a given weight threshold provides the final solution to the localization. A target node’s final computed location is expressed as a geographic region bounded by straight-line segments. Straight-line approximations represent areas concisely, expediting arithmetic operations on these areas.

Localizing a target to a region, rather than a point, conveys information that cannot be expressed from single point geolocalization in that it shows every possible point at which a target may be located. In some cases, a target node may be located within any of several disjoint regions, whose nature cannot be conveyed from information of a single point estimate. Nonetheless, a single point estimate may be extracted from the localization region for comparison to other geolocalization techniques.

Alidade provides single-point estimates of target localizations using a Monte Carlo technique that calculates the center of mass point within the localization region. For a disjoint localization region, it chooses a point inside the region that is closest to the center of mass. This is done in order to support legacy applications that require single-point location estimates of targets, as well as to aid in evaluating the accuracy characteristics of the system.

2.5 Distributed Framework

Previously, the Octant framework originally made use of a single core on a single machine. This implementation was acceptable since the scope was within the continental U.S., which limited the size and number of localizations. However, this approach is infeasible for global-scale geolocalization, in which the number of landmarks, beacons, and targets are orders of magnitude greater than it was previously. Thus, runtime would be astronomical unless parallelism is brought into the picture. In order to improve performance, Alidade bypasses limitations on memory and CPU on a single processor, and instead leverages the availability of cheap commodity machines.

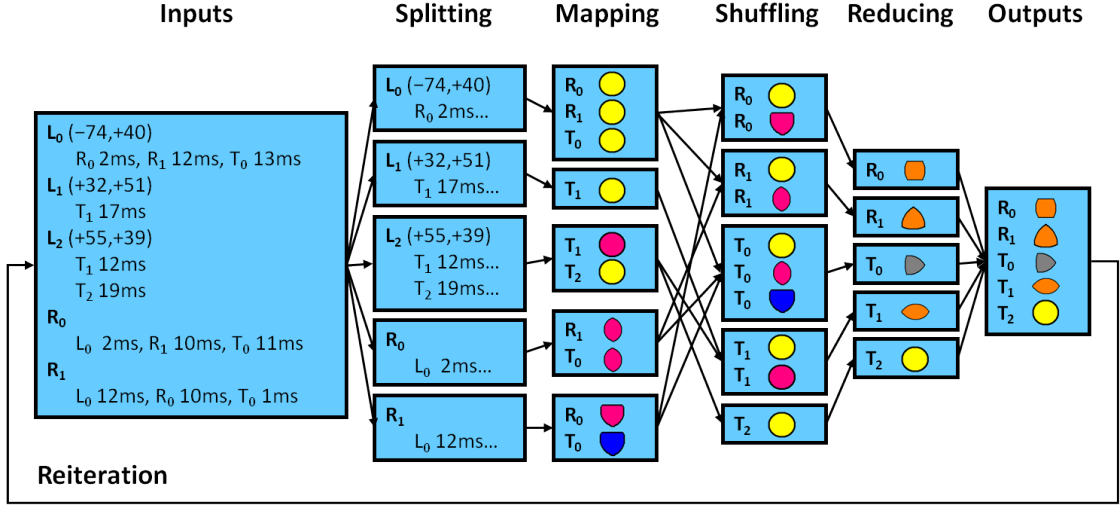


Figure 2.8: MapReduce process for target localization.

Alidade achieves high-performance and scalability through a radically new design for geolocation systems. The system massively parallelizes both the constraint extraction and evaluation by partitioning the input and intermediate data into independent sets that can be concurrently executed. Such a partitioning enables a MapReduce [8] style implementation that scales well with the number of landmarks and targets in the system, as well as the number of available cluster nodes. Because MapReduce is a popular parallelism technique, the framework has a standard interface for which a number of commercial clusters are available. The overall MapReduce structure is illustrated in Figure 2.8.

The Map stage reads input data derived from traceroutes between landmarks and targets. The data is partitioned among landmarks, where each partition maintains a landmarks latencies to targets and to intermediate nodes encountered on traceroutes to the targets. The Map stage extracts latency-based geometric constraints for each target listed in the partition. The output of the

Map stage is then sorted and divided into new partitions for the Reduce stage, where each partition contains all constraints for a target. The Reduce stage localizes each target and intermediate node based on the constraints in the partition. This MapReduce process can be repeated to further refine localizations, using output data from the previous process, where each successive iteration converges to optimal localizations.

CHAPTER 3

FURTHER ENHANCEMENTS OF FRAMEWORK

In addition to the key concepts that are fundamental to Alidade, many other techniques that improve localization accuracy and performance come into play.

3.1 Intermediate Node Localization

Therefore, beacons provide more precise router localizations, and these router localizations improve target localizations. To demonstrate, suppose the objective is to localize the target (T). Normally, a set of active landmarks (L1, L2, L3) traceroute to the target, as shown in Figure 3.1(a). In addition, the active landmarks traceroute to beacons (B1, B2, B3), as shown in Figure 3.1(b). Alidade uses active landmarks and beacons to localize routers shared on multiple traceroutes (R1, R2). These routers in turn localize the target. Finally, the result is a more precise target localization.

3.2 Intermediate Node Latencies

Extracting constraints from network latency measurements is non-trivial in that latencies are often erratic and dilated due to network factors such as circuitous routing. To mitigate the effects of circuitous routing, Alidade uses interhop latencies and connectivity information to localize intermediate nodes on traceroutes between landmarks and targets. Alidade can efficiently use a node's feasibility region to further refine the estimated locations of targets, and thus can use intermediate nodes as landmarks with approximately known locations. De-

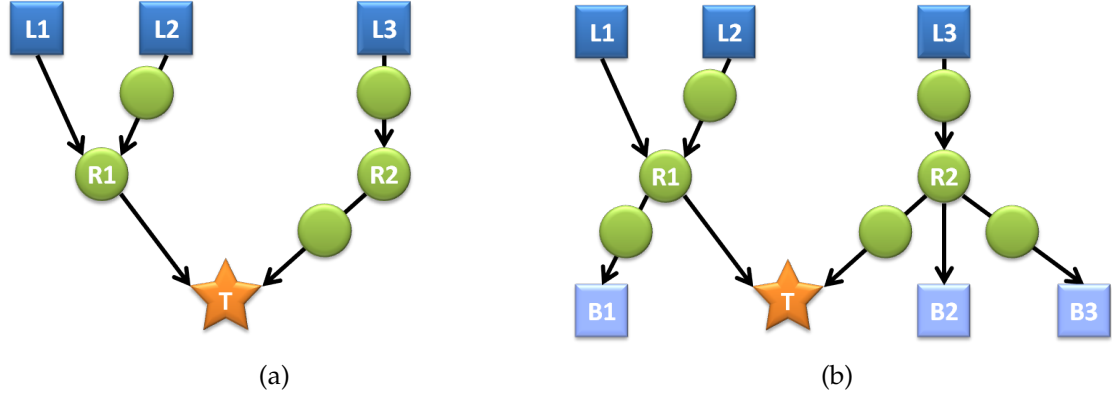


Figure 3.1: Beacon demonstration. (a) Active landmarks (L1, L2, L3) traceroute to the target (T). (b) Active landmarks traceroute to beacons (B1, B2, B3).

creasing the granularity of the geolocalization aids in factoring out the indirect paths that would otherwise render large and imprecise areas.

Extracting constraints from interhop latencies still poses a challenge in that varying network paths, software delays, and overloaded routers affect latency. Indeed, indirect latencies can lead to underestimation issues and even negative values. This occurs because indirect latencies are not measured directly, and instead are obtained by subtracting one measured latency from another, as shown in Figure 3.2(a). Some latency measurements are inflated enough to cause negative indirect latencies, as shown in Figure 3.2(b). To resolve these issues, the system applies a number of techniques. The system ensures that latency monotonically increases with hop count, which removes negative latencies, as shown in Figure 3.2(c). However, even if a latency is not negative, it may still be underestimated. Thus, Alidade can increase small latencies to a specified minimum value (e.g. 1.0 ms), multiply latencies by a specified amount (e.g. 1.15), or add to latencies with a fixed amount (e.g. 2.0 ms).

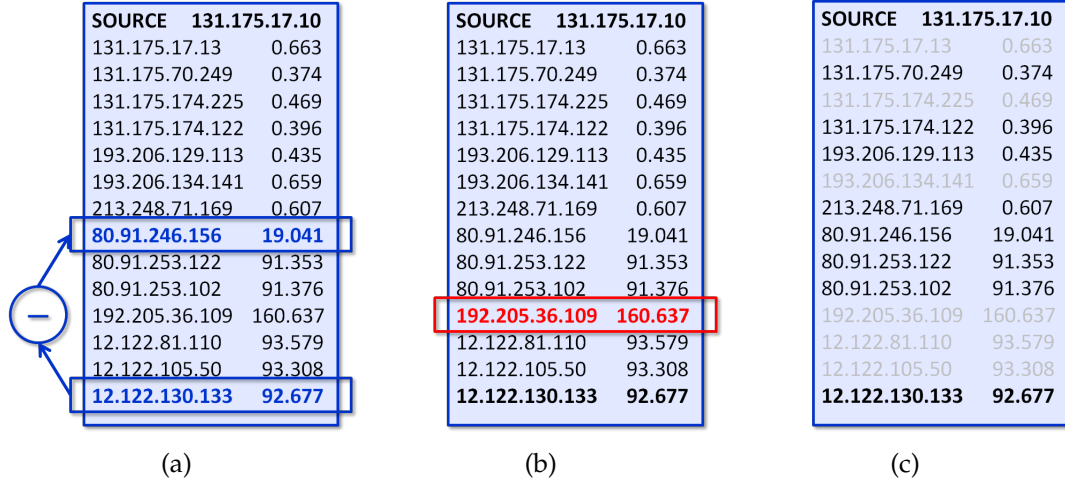


Figure 3.2: Traceroute example. (a) Indirect latency obtained from subtraction. (b) Dilated latency that causes a negative indirect latency. (c) Monotonically increasing traceroute data.

3.3 Public Host Extraction

To take advantage of these potential landmarks, the Bing database was mined for public hosts. After issuing 1,005,000 queries across the globe in a 10 km x 10 km grid, 228,204 public institutions with websites were found. At the time of writing, Bing provides information for businesses, schools, and libraries in the United States, the United Kingdom, Germany, Spain, and France.

However, Alidade cannot use all of public hosts as beacons. Many small businesses host websites at third party hosting services and large businesses expand to multiple locations and host websites at offsite locations. Therefore, Alidade populates two sets of potential landmarks that satisfy certain constraints.

The first set of landmark candidates contains businesses that are residentially hosted. In other words, each business has a reverse DNS name that

Localization Accuracy of Public Hosts		
	Correct	Incorrect
Residential	547	49
Self-Hosted	2129	2800
Self-Hosted BL	1996	0

Figure 3.3: Localization accuracy of public hosts.

matches a residential IP format (e.g. brownpublishing.com maps to rrccs-70-60-47-71.central.biz.rr.com). However, websites such as rackspace.com or Amazon EC2 (Elastic Compute Cloud) still remain. To remove these results, the system uses a white list of 47 Internet service providers. The system extracted 2,916 residential hosts (14,656 prior to filtering with the white list).

The second set of landmark candidates contains businesses that are self-hosted. In other words, each business has a reverse DNS hostname that matches the original hostname (e.g. petersheatingandair.com maps to www.petersheatingandair.com). However, websites such as staples.com and walmart.com still remain. To remove these results, the system uses a blacklist of /24s, discussed below. The system extracted 1,996 self-hosted domains (19,990 prior to filtering with the blacklist).

Finally, the filtered public host sets were crosschecked with other sources, including Speed-of-light constraints and the MaxMind database. To crosscheck with Speed-of-Light constraints, traceroutes were conducted from PlanetLab nodes to the public hosts. Alidade localized the public hosts using Speed-of-Light constraints from PlanetLab landmarks to ensure that the public hosts exist within a reasonable distance of their Bing-based locations. Figure 3.3 shows the number of times the Bing-based locations fall inside ("Correct") and outside ("Incorrect") their localization regions. For the residential host set, the Bing-

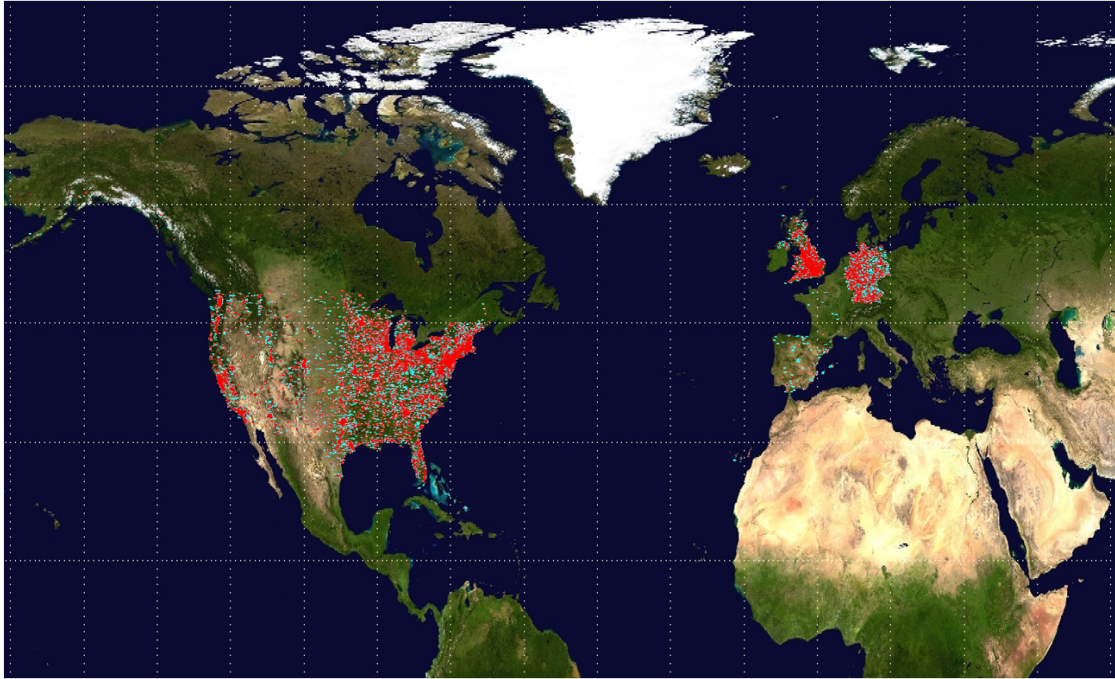


Figure 3.4: Distribution of public host locations.

based locations were inside CBG region over 90% of the time. For the self-hosted set, a blacklist of /24s was developed that included IP addresses associated with “Incorrect” localizations. Figure 3.4 illustrates the distribution of the remaining public hosts.

3.4 Additional Geolocalization Tools

Recall that Alidade can create landmarks with approximately known locations from routers. Additional geolocalization tools can provide feasibility regions for these routers so that Alidade can use the latter as landmarks. UNDNS and HostParser each consist of a set of regular expressions that map a given reverse

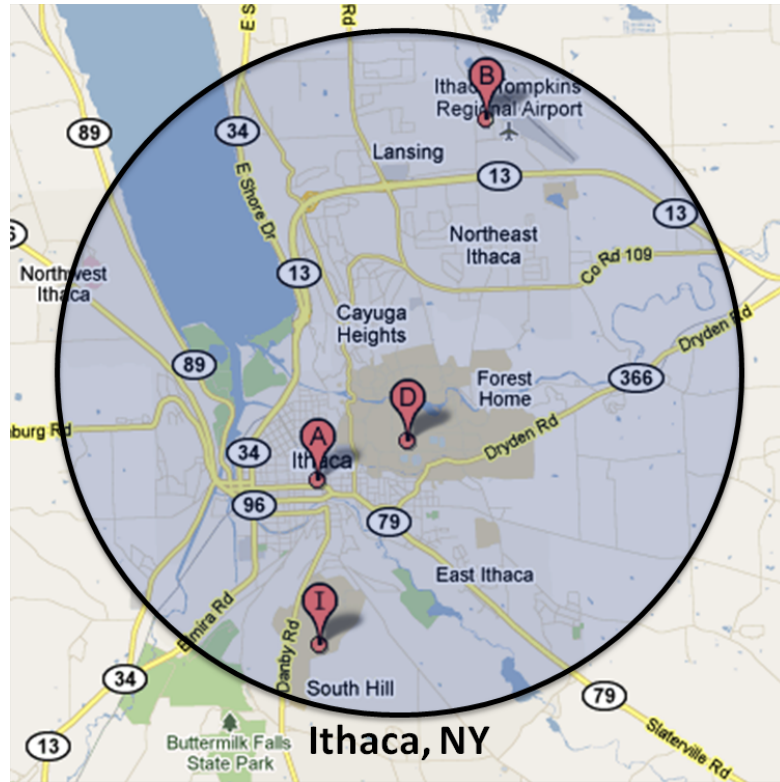


Figure 3.5: City name to region mapping in Octant.

DNS name to a city name. A reverse DNS name is often structured in that it contains an airport code or city abbreviation. Alidade extracts the set of all IP addresses from traceroutes, performs a reverse DNS lookup on each IP address, and queries UNDNS via a MapReduce process or submits a set of IPs to Host-Parser. For each IP, these tools return a city where the associated node may be located. Given that Alidade can map these names to regions, the associated nodes can be used as landmarks with approximately known locations.

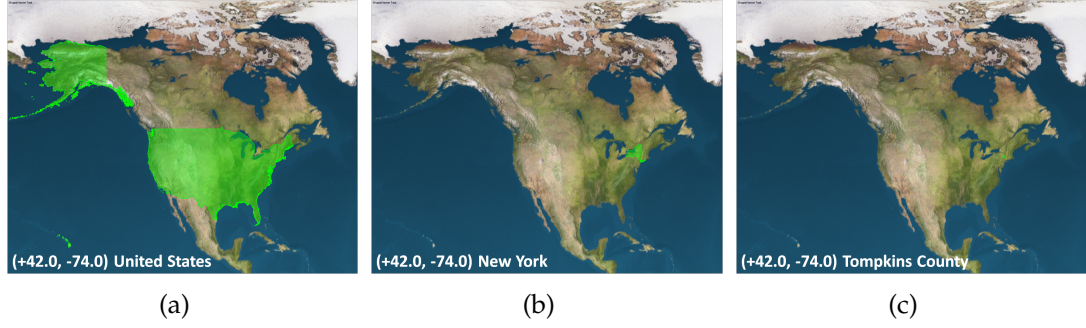


Figure 3.6: City name to region mapping in Alidade for the United States. (a) National level. (b) State level. (c) County level.

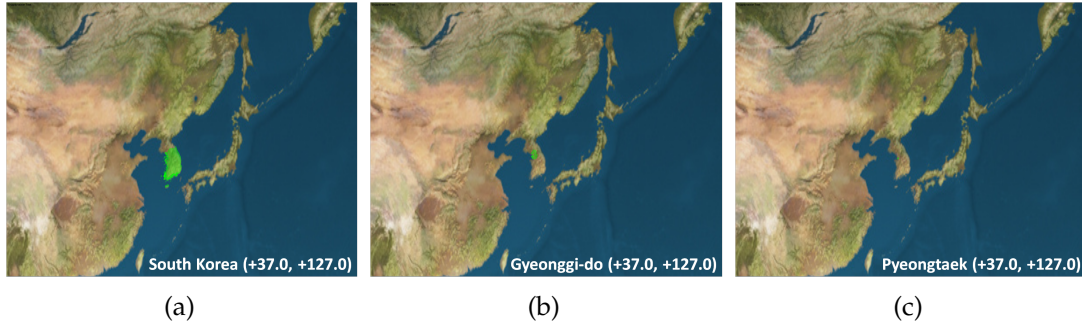


Figure 3.7: City name to region mapping in Alidade for South Korea. (a) National level. (b) Provincial level. (c) Municipal City level.

3.5 Geopolitical Boundaries

Octant mapped a city to a region by finding the zip code locations in the city, drawing circles centered on the zip code locations, and constructing a bounding circle around these areas. Figure 3.5 illustrates this process. However, this approach is very approximate and does not easily scale to other nations whose zip code information is not readily available.

Alidade maps a city to a precise geopolitical boundary constructed from Esri Shapefiles [2] that are publicly available [3]. A shapefile is a standard geospa-

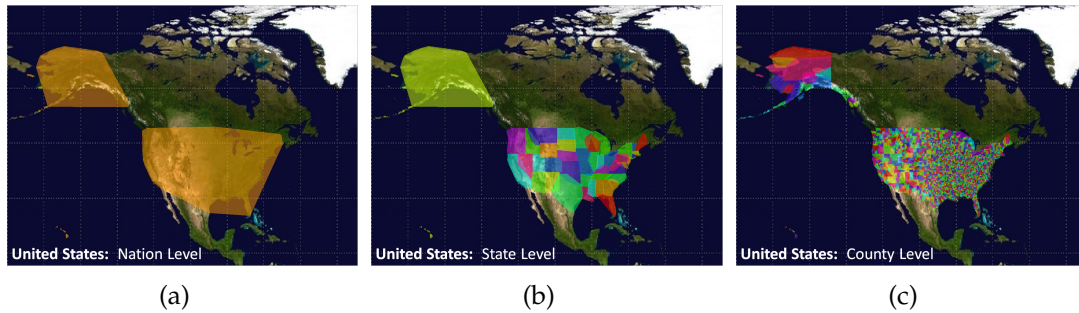


Figure 3.8: Convex hull representation of geopolitical boundaries. (a) National level. (b) State level. (c) County level.

tial vector data format that many geographic software applications use. Alidade uses worldwide data for level 0 (national), level 1 (equivalent to state level), and level 2 (equivalent to county level) administrative boundaries. Alidade uses a search method that maps a city name to a coordinate and returns the administrative area that contains the coordinate. This search primitive provides more flexibility in that it can locate any shapefile entry regardless of knowledge of region names and administrative level names, which vary among nations.

To efficiently find a region, Alidade constructs a grid that covers the globe and partitions it into squares that cover small increments. To each grid square, the system assigns a list of shapes that intersect it. When given a point query, the system finds the grid containing the point and fetches the small subset of shapes associated with it. Figure 3.6 and Figure 3.7 illustrate two examples of this search primitive.

Each administrative region can have up to 7,430 rings and a ring can have up to 724,482 points. Expanding a shape of this complexity takes an enormously long time. Therefore, Alidade takes the convex hull of the rings. Expanding a convex hull shape with 10s of points is much more efficient and is relatively

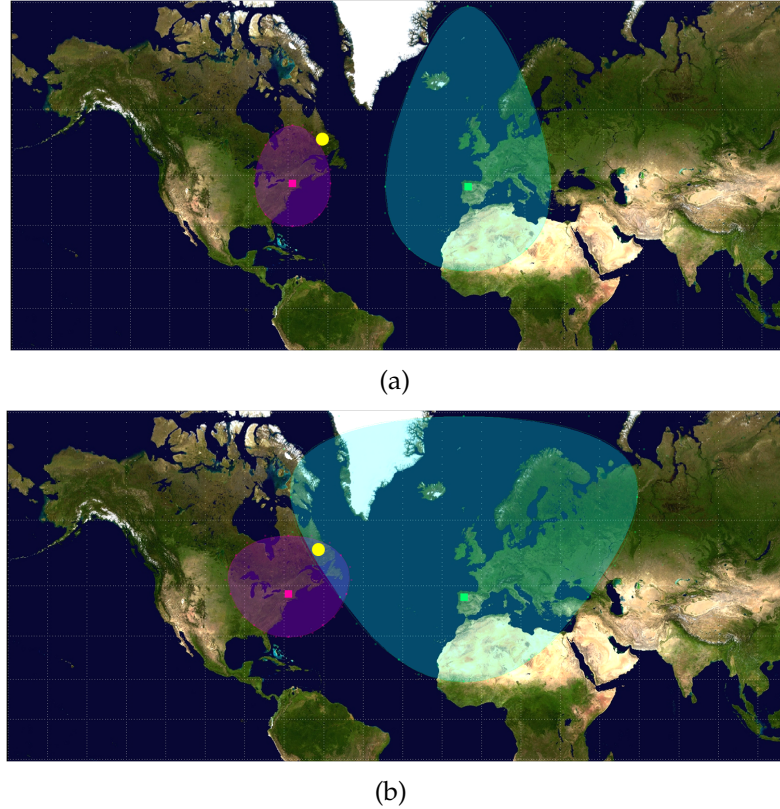


Figure 3.9: Flat world and geodesic-aware approximations on a Mercator map. (a) Localization of a target (circle) on a flat world. (b) Localization of the same target using geodesic-aware projection.

accurate given the size of the original shape. Figure 3.8 illustrates an example of this technique for the United States.

3.6 Geodesic-Aware Projection

Previously, Octant approximated the world to a flat surface, illustrated in Figure 3.9(a), which was acceptable since the scope was limited to the continental United States. This ensured that localizations were sufficiently small and far from the poles, thus limiting distortion. In order to support worldwide localiza-

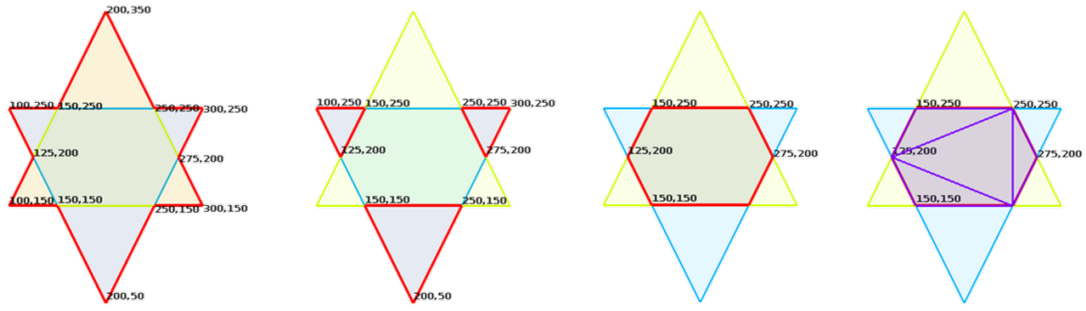


Figure 3.10: Efficient area arithmetic for addition, subtraction, intersection, and triangulation.

tion, Alidade takes into account the spherical nature of the globe by employing geodesic-aware projection that approximates the Earth to a sphere, illustrated in Figure 3.9(b). Geodesic region representation is based on the distance each point is from the equator and from the center of the area. Simulations have shown that the localization accuracy improves significantly, especially for larger regions.

To demonstrate the effect of this approach on localization accuracy, suppose that there exists a target in Newfoundland, a landmark in Maine that pings the target at 30 ms, and a landmark in Santiago that pings the target at 100 ms. A flat world approach would generate circular areas (note that they are drawn on a Mercator map) that incorrectly do not encompass target. On the other hand, geodesic-aware regions have drastically different and more realistic shapes, which improve localization accuracy.

3.7 Region Representation

Octant generated areas with mathematically complex cubic Bézier curves. The Bézier curves did not properly expand geodesically and consumed excessive runtime. To address these issues, Alidade uses a simplified area representation that replaces Bézier curves with more efficient straight-line approximations. This is a compact and precise method that significantly reduces memory and runtime consumption. It supports complex regions that may be non-convex, be disconnected, or contain holes. This allows any kind of constraint to be represented, enabling the system to reason about both positive constraints and negative constraints. Alidade performs efficient manipulation of a vast shape space. The straight-line approximation enables quick and space-efficient computation of geometric intersection, union, subtraction.

Since a majority of the memory and runtime consumption still takes place in area manipulations, we devised an area arithmetic tool (addition, subtraction, intersection, triangulation) that we optimized for polygons, illustrated in Figure 3.10. To further improve efficiency, we plan to move away from Javas library Area implementation to this more efficient version.

3.8 Region Expansion

In order to dilate the feasibility region of an approximately known landmark based on latency information, the system uses a region expansion technique. Octant performed complex area expansion, illustrated in Figure 3.11(a), by taking the union of the original shape and numerous circles distributed along an

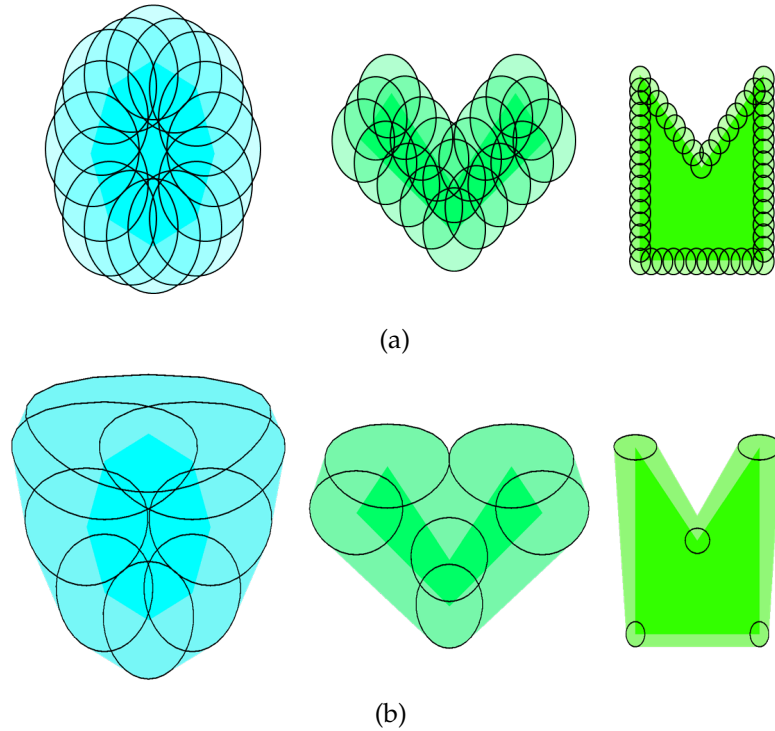


Figure 3.11: Area expansion techniques. (a) Octant's implementation. (b) Alidade's implementation.

area's boundary. The resulting shape was complicated, having many semicircles along its boundary, and accuracy depended heavily on the concentration of the circles along the path. As a result, execution suffered from numerous areas and arithmetic operations. The only way to expand an area without the computational expense was to approximate the area with a circle and increase the radius. However, this is at the expense of precision. Finally, region expansions did not follow geodesic behavior.

Alidade introduces an accurate and efficient geometric expansion technique, illustrated in Figure 3.11(b), that iterates across an area's inflection points and creates circles of desired expansion size centered on these points. It then finds the convex hull of each pair of adjacent circles. The final shape is the union of the

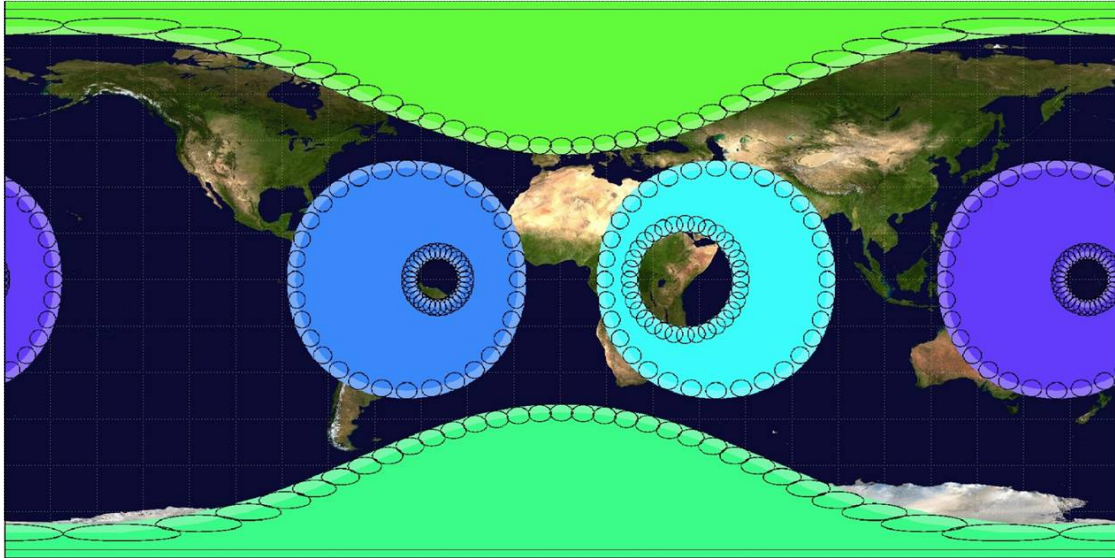


Figure 3.12: Examples cases for area expansion in Alidade.

original shape and convex hulls. This new technique maintains accuracy with far fewer areas and arithmetic operations and handles irregular areas, including those with holes. Region expansions also incorporate geodesic behavior.

3.9 Region Boundary Crossing

Handling areas that cross the International Date Line and poles was not necessary in Octant since all nodes were restricted to the continental United States. To handle these cases, Alidade represents areas differently. Rather than having multiple special cases, the new implementation is much simpler and less bug prone. It allows a shape to naturally cross boundaries, copies the shape twice at $\pm 360.0^\circ$ longitude, and finally intersects all copies with a bounding rectangle with corners at $(-90.0^\circ, -180.0^\circ)$, $(-90.0^\circ, +180.0^\circ)$, $(+90.0^\circ, -180.0^\circ)$, and $(+90.0^\circ,$

+180.0°). For a region that overlaps a pole, a hole incorrectly forms close to the pole. The size of this hole depends on the extent that the region overlaps the pole. To solve this issue, the system automatically completes the region with a simple rectangle. This rectangle covers the pole to the necessary extent and spans $\pm 180.0^\circ$ longitude. Figure 3.12 illustrates several test cases that demonstrate the accuracy of both the expansion code and area representation code.

CHAPTER 4

EVALUATION

Alidade used PlanetLab, a shared research test bed with hundreds of nodes distributed across the globe, to provide a set of active landmark nodes. A large-scale map of the Internet was constructed, with Akamai and Bing Phonebook beacons as mock targets. The system used network latency measurements and connectivity information from 16 sets of traceroutes collected over a two month period. In each set, traceroutes were sent from and to 329 unique PlanetLab locations and reached 436 unique beacon locations. This resulted in 251,685 unique paths and a total of 4.0 million traceroutes.

To make full use of the MapReduce framework, Alidade runs on a reserved sixteen-node computing cluster. Each node contains eight Xeon microprocessor cores that run at 2.5 Gigahertz and is equipped with 16 Gigabytes of RAM. Alidade is deployed on the Hadoop implementation of MapReduce.

Alidade evaluates several geolocalization metrics, including accuracy, precision, and correctness. We coalesce PlanetLab hosts at the same University and Akamai servers at the same Ecor center. Not only does this reduce load on the cluster, but it also ensures that Alidade does not localize targets with routers that have been, in turn, localized by beacons at the same data center. We partition the set of coalesced Akamai beacons into 12 subsets. The system then batch localizes a subset of 35 Akamai beacons and uses the remaining ones as passive landmarks.

The system evaluates the accuracy of localizations by calculating the error of each target's single point estimate with respect to its actual location. Alidade

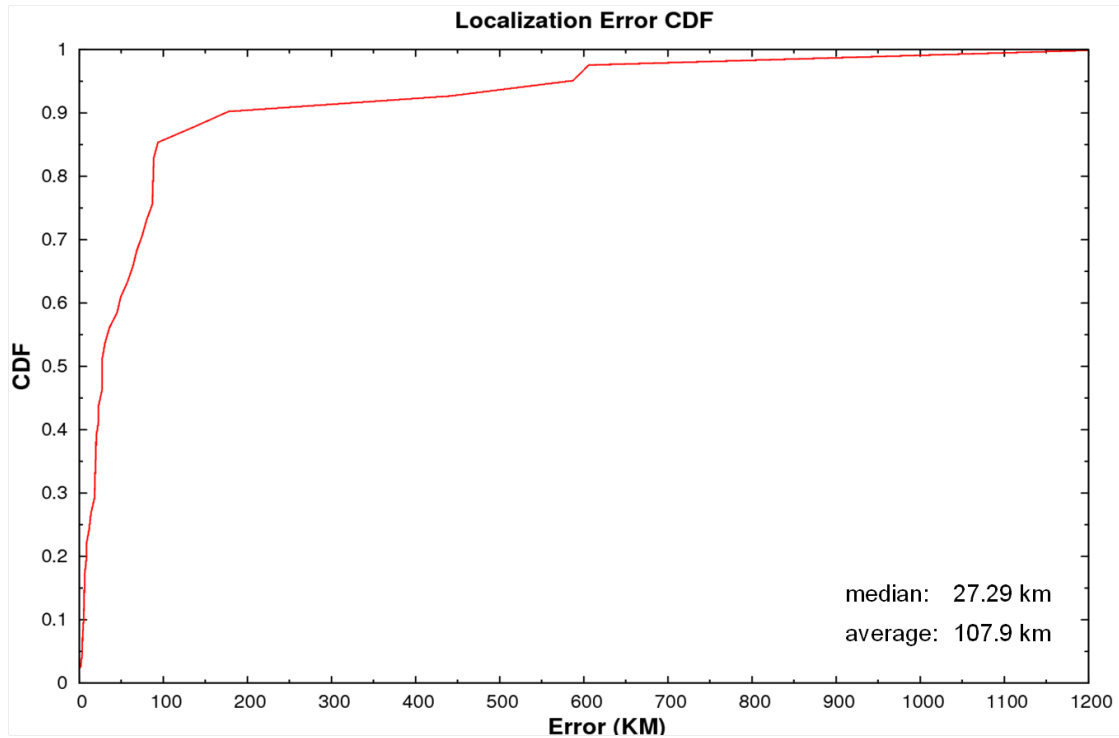


Figure 4.1: Evaluation of localization accuracy.

can localize targets distributed worldwide with a median error of 27.29 km and average error of 107.9 km. Octant can localize targets limited to the continental United State with a median error of 36 km. Figure 4.1 shows the CDF of accuracy results.

Alidade evaluates the precision of localizations by calculating the size of each target's localization region. Alidade can localize targets distributed worldwide to a median region radius of 119.8 km and average region radius of 403.8 km. Octant can localize targets limited to the continental United State to a median region radius of 301 km. Figure 4.2 shows the CDF of precision results.

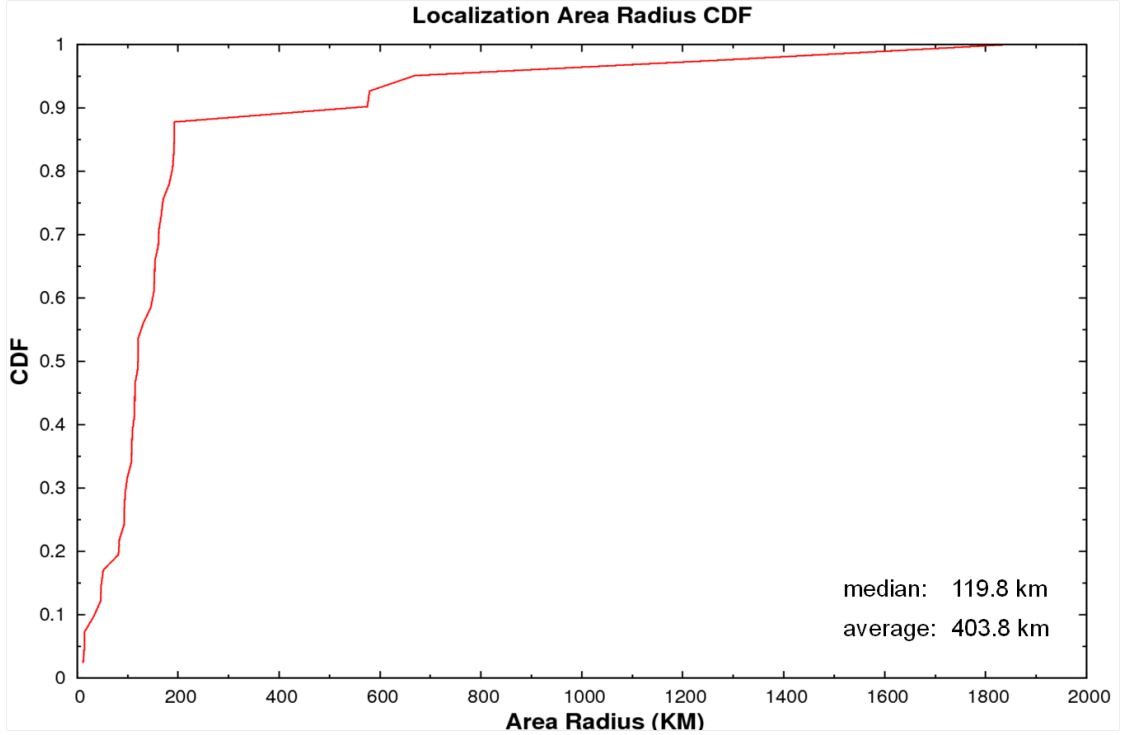


Figure 4.2: Evaluation of localization precision.

The system evaluates the correctness of localizations by determining whether each localization region contains the actual location of the target. The correctness result is 100%; every localization region contains the target.

Finally, Alidade evaluates the effects of various components on the above localization metrics. Figure 4.3 shows the progression of results as these components are introduced individually. "Minimum Cutoff Latency" describes the minimum value to which Alidade increases smaller latencies. "Fraction of Weighted Layers" describes how Alidade chooses the layers to include in the localization region of each target, where "max weight" is the maximum weight found across all layers in a target's weighted intersection. "Latency Slack Multiplier" describes the value by which Alidade multiplies latencies. Of all the

Progression of Results			
Introduced Component	Median Error	Average Error	False Count
Active Landmarks <i>PlanetLab hosts</i>	186.49	485.72	0
Passive Landmarks <i>Beacons & Routers</i>	33.21	109.59	8
Minimum Cutoff Latency <i>1 ms</i>	41.28	119.16	0
UNDNS	42.89	123.13	1
Fraction of Weighted Layers <i>95% max weight</i>	28.57	110.83	0
Latency Slack Multiplier <i>125%</i>	27.29	107.90	0

Figure 4.3: Evaluation of the introduction of individual components on localization metrics.

features evaluated, the introduction of passive landmarks makes the most significant positive impact on the localization metrics. Defining the fraction of weighted layers to be those with 95% of the maximum weight considerably improves the accuracy and correctness results.

CHAPTER 5

RELATED WORK

The geographic localization of internet hosts can be separated into two main categories: single-point geolocalization and region geolocalization.

5.1 Single Point Geolocalizaion

A number of geolocalization techniques estimate the locations of nodes via single point representations. IP2Geo [13] offers three geolocalization services: GeoPing, GeoTrack, and GeoCluster. Other services such as NetGeo, IP2LL, Quova, Gtrace, and Visual-Route also provide single point geolocalization.

GeoPing maps the target to the landmark that has the closest latency characteristics. These latency characteristics are based on similarities among network signatures [7]. GeoPing's accuracy depends on the number of landmarks as well as how close these landmarks are to the targets.

GeoTrack performs a traceroute to a given target, extracts geographical information from the DNS names of routers on the path, and localizes the node relative to the last router on the path whose position is known. GeoTrack's accuracy depends on the distance between the last recognizable router and the landmark, as well as the reliability of router name to location mapping.

GeoCluster breaks the IP address space into clusters of nodes that are likely to be in the same location, and then assigns a location to each cluster based on IP-to-location mappings from third-party databases. These databases include the user registration records from a large email service and the zip codes of

users from an online TV program guide. GeoCluster’s accuracy requires a large, fine-grain database that is constantly refreshed. Such databases are not readily available to the public due to potential privacy concerns. ISPs may perform clustering inaccurately when assigning IP address ranges, and the clustering itself may not sufficiently capture locality.

NetGeo [12] and IP2LL [12] use the locations recorded in the WHOIS database for the corresponding IP address block to localize nodes. NetGeo and IP2LL’s accuracy requires that the IP address blocks are not geographically diverse and that the WHOIS database is accurate. However, the WHOIS database is not closely regulated and it often points to the location of the owner’s head office, which is not always within the vicinity of the actual target.

Quova [4] is a commercial service that localizes nodes based on a proprietary technique. Neither the details of the technique nor a sample data set is publicly available. Gtrace [14] localizes intermediate and endpoint nodes by using NetGeo, DNS LOC (a database that maps domain names to locations), and domain name country codes. Visual-Route [5] is a commercial traceroute tool that localizes intermediate and endpoint nodes.

5.2 Region Geolocalization

A number of geolocalization techniques, such as GeoLim [9], Topology-Based Geolocation (TBG) [11], and Hop-Based Geolocation (HBG) [6] estimate the locations of nodes via region representations, which describe where the nodes are likely to exist.

GeoLim [9] derives the estimated position of a node via landmark-to-target latency measurements, and extracts upper bounds on region representation sizes based on inter-landmark latency-to-distance ratios. GeoLim then localizes the node in the region formed by the intersection of these areas relative to established landmarks. GeoLim does not use negative information, does not permit non-convex regions, and does not handle uncertainty. Thus, the accuracy and precision of this technique breaks down as inter-landmark distances increase.

Alidade, in contrast, combines both positive and negative constraints to yield a small bounded region that contains the node’s location. More specifically, it uses negative information for localization, selects a single-point estimate from the set of points in which a node might be located, permits areas based on non-convex latency-to-distance models, and aggressively fine tunes constraints from network latency measurements.

Topology-Based Geolocation (TBG) [11] uses the maximum transmission speed of packets in fiber to conservatively determine the region where the target lies. TBG uses router latencies on the landmark-to-target paths to place the routers and targets in such a way that minimizes inconsistencies with the latencies. TBG relies on a global optimization scheme that minimizes the average position error for both routers and targets. This methodology can introduce excessive errors in the target position in order to reduce errors in the intermediate router positions.

Alidade, in contrast, does not strive to achieve global optimization, but instead provides a much simpler geometric solution. Alidade performs geolocation in near real-time, while TBG requires significantly more computation time and resources. Additionally, Alidade seamlessly incorporates more

sources of information to produce exogenous geometric constraints. These additional sources of information include geographical boundaries and demographical information, such as population density.

Hop-Based Geolocation (HBG) [6] uses the number of hops from landmarks to targets to perform geolocalization. HBG avoids using inter-latency information from traceroute measurements and instead assumes that the routers are evenly spaced in a straight line between landmark and target. All of the estimates derived from multiple traceroute measurements to a target are averaged to obtain a final estimate of the target's location.

Alidade, in contrast, uses raw network latency measurements instead of assuming that routers are evenly distributed, as the latter often leads to severe inaccuracies. This is especially the case for longer network paths.

5.3 Wireless Geolocalization

The most comprehensive work in geolocalization in wireless networks is Sextant [10]. Sextant and Alidade share several geolocalization techniques. They use both positive and negative constraints and allow nodes whose positions are not fully known to localize other nodes.

Alidade differs substantially from Sextant in several ways. Sextant maps successful packet transmissions to distance constraints, while Alidade maps latencies to distance constraints. In Sextant, indirect routes are not an issue, since all wireless network nodes serves as both routers and end hosts.

CHAPTER 6

CONCLUSION

Alidade is a distributed framework for accurate and scalable geolocalization with worldwide coverage. The system leverages extensive data from many diverse sources of ground truth, including the PlanetLab testbed, a global-scale dataset of Ecor web servers from Akamai Technologies, a public host dataset, and landmarks whose positions are only approximately known. In addition, the system integrates other geolocalization tools, including UNDNS and Host-Parser. In order to scale globally, the system uses a compact geometric region representation that conserves memory and runtime. To efficiently handle large-scale datasets and network measurements, Alidade is structured as a distributed framework that supports large-scale parallel multistage geolocalization. Alidade surpasses its predecessor, Octant in all evaluated localization metrics. The worldwide geolocalization that Alidade accomplishes in less than ten minutes does not reach completion on the Octant framework.

BIBLIOGRAPHY

- [1] EdgeScape. <http://www.akamai.com/html/technology/products/edgescape.html>.
- [2] Esri Shapefile. <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.
- [3] Global Administrative Areas. <http://www.gadm.org/world>.
- [4] Quova. <http://www.quova.com>.
- [5] Visualware Inc. <http://www.visualroute.com>.
- [6] R. Baden. IP Geolocation in Metropolitan Area Networks. <http://www.cs.umd.edu/Grad/scholarlypapers/papers/RandolphBaden.pdf>.
- [7] P. Bahl and V. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM*, Tel Aviv, Israel, March 2000.
- [8] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *Operating Systems Design and Implementation*, San Francisco, CA, December 2004.
- [9] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-Based Geolocation of Internet Hosts. In *Internet Measurement Conference*, Taormina, Sicily, Italy, October 2004.
- [10] S. Guha, R. Murty, and E.G. Sirer. Sextant: A Unified Framework for Node and Event Localization in Sensor Networks. In *Mobihoc*, Urbana-Champaign, IL, May 2005.
- [11] E. Katz-Bassett, J. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. Towards IP Geolocation using Delay and Topology Measurements. In *Internet Measurement Conference*, Rio de Janeiro, Brazil, October 2006.
- [12] D. Moore, R. Periakaruppan, and J. Donohoe. Where in the World is net-geo.caida.org? In *INET2000 Poster*, Yokohama, Japan, July 2000.
- [13] V. Padmanabhan and L. Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *SIGCOMM*, San Diego, CA, August 2001.

- [14] R. Periakaruppan and E. Nemeth. GTrace - A Graphical Traceroute Tool. In *LISA*, Seattle, WA, November 1999.
- [15] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *SIGCOMM*, Pittsburgh, PA, August 2002.
- [16] B. Wong, I. Stoyanov, and E. G. Sirer. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. In *Networked System Design and Implementation*, Cambridge, MA, April 2007.
- [17] Inja Youn, Brian L. Mark, and Dana Richards. Statistical Geolocation of Internet Hosts. In *18th International Conference on Computer Communications and Networks*, San Francisco, CA, August 2009.